Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

Claim 1. (Original) An integrated circuit comprising:

a graphics pipeline; and

a frame buffer interface,

wherein the graphics pipeline comprises a shader coupled to a texture cache, the shader further coupled to the frame buffer interface,

wherein the shader stores and loads data to and from an external graphics memory using the frame buffer interface, and the shader is configured to store and load data to and from specific locations in the external graphics memory during a single pass through the graphics pipeline.

Claim 2. (Original) The integrated circuit of claim 1 wherein the shader executes instructions forming a shader program, the shader program executed during a plurality of passes.

Claim 3. (Original) The integrated circuit of claim 2 wherein one of the plurality of passes comprises:

executing a first plurality of instructions;

executing a read command, wherein data is read from a buffer in the external graphics memory and received by the shader using the frame buffer interface; and

executing a second plurality of instructions.

Claim 4. (Original) The integrated circuit of claim 2 wherein one of the plurality of passes comprises:

executing a first plurality of instructions;

executing a write command, wherein data is written to a buffer in the external graphics memory by the shader using the frame buffer interface; and executing a second plurality of instructions.

Claim 5. (Currently amended) The integrated circuit of claim 1 wherein the shader ~~may~~ is configured to write to and read from a first number of buffers located in the external graphics memory, the first number greater than two.

Claim 6. (Original) The integrated circuit of claim 3 wherein the first number of buffers are read-only buffers.

Claim 7. (Original) The integrated circuit of claim 5 wherein the first number of buffers are read/write buffers.

Claim 8. (Original) A method of generating a computer graphics image comprising:

executing a first plurality of instructions in a shader program, the shader program running in a shader in a graphics pipeline, the shader program executed during a plurality of passes through the shader;

executing a load command, wherein data is read from a first buffer by the shader; and

executing a second plurality of instructions in the shader program,

wherein the first plurality of instructions, the load command, and the second plurality of instructions are executed during a single pass through the shader.

Claim 9. (Currently amended) The method of claim 8 wherein the shader ~~may~~ is configured to load data from a plurality of buffers, the plurality of buffers exceeding two buffers.

Claim 10. (Original) The method of claim 8 further comprising:

executing a store command, wherein data is stored by the shader in a second buffer; and

executing a third plurality of instructions in the shader program, wherein the first plurality of instructions, the load command, the second plurality of instructions, the store command, and the third plurality of instructions are executed during a single pass through the shader.

Claim 11. (Currently amended) The method of claim 10 wherein the shader ~~may~~ is configured to store data in and load data from a plurality of buffers, the plurality of buffers exceeding two buffers.

Claim 12. (Original) The method of claim 11 wherein the buffers are located in an external graphics memory.

Claim 13. (Original) The method of claim 12 wherein the shader stores data in and loads data from the buffers using a frame buffer interface.

Claims 14.-20. (Cancelled)

Claim 21. (New) A method of generating a graphics image using an integrated circuit, the integrated circuit comprising:

a frame buffer interface configured to read data from and write data to a graphics memory; and

a graphics pipeline comprising a shader, the shader coupled to the frame buffer interface;

the method comprising:

executing a shader program using the shader, the shader program executed during a plurality of passes, one of the plurality of passes comprising:

executing a first plurality of instructions;

executing a write command, wherein data is written to a first buffer in the graphics memory by the shader using the frame buffer interface; and

executing a second plurality of instructions.

Claim 22. (New) The method of claim 21 wherein the shader is configured to access a plurality of buffers in the graphics memory using the frame buffer interface.

Claim 23. (New) The method of claim 22 wherein the shader is coupled to a texture cache.

Claim 24. (New) The method of claim 21 further comprising:

executing a read command, wherein data is read from a second buffer in the graphics memory by the shader using the frame buffer interface; and

executing a third plurality of instructions.

Claim 25. (New) The method of claim 24 wherein the read and write commands in the shader program include storage location identifications.

Claim 26. (New) The method of claim 24 wherein read and write commands in the shader program include a direct reference to storage location addresses in the graphics memory.

Claim 27. (New) The method of claim 24 wherein read and write commands in the shader program include indirect references to storage location addresses in the graphics memory.